



Joel Syrjämäki

SPOCO-RANNETIETOKONEEN TOISEN SUKUPOLVEN TOTEUTUS

SPOCO-RANNETIETOKONEEN TOISEN SUKUPOLVEN TOTEUTUS

Joel Syrjämäki
Opinnäytetyö
Kevät 2012
Tietotekniikan koulutusohjelma
Oulun seudun ammattikorkeakoulu

TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu
Tietotekniikan koulutusohjelma, elektroniikan suunnittelu ja testaus

Tekijä: Joel Syrjämäki

Opinnäytetyön nimi: SpoCo-rannetietokoneen toisen sukupolven toteutus

Työn ohjaaja: Ensio Sieppi

Työn valmistumislukukausi ja -vuosi: Kevät 2012

Sivumäärä: 40 + 5 liitettä

Opinnäytetyön tavoitteena oli suunnitella ja kehittää rannetietokoneen laitteistoa ja ohjelmistoa SpoCo-projektin vaatimuksien mukaisesti. SpoCo-rannetietokoneen kehityksessä oli mukana fysioterapia-alan ammattilaisia ja Lapin yliopiston sekä Jyväskylän yliopiston henkilöitä. Projektin tavoitteena oli kehittää rannetietokoneen prototyyppiä palvelemaan kuntosaliharjoittelijoita. Rannetietokoneen tarkoituksena oli mitata kuntosaliliikkeitä ja sen mukaan kertoa käyttäjälle kuntosalilla tehtyjen liikkeiden toistomäärä. Tarkoituksena oli saada markkinoille rannetietokone, joka on suunnattu kuntosaleilla käyville asiakkaille.

Kehitysalustana käytettiin Arduino Atmega1280-elektronikka-alustaa. Muu rannetietokoneen laitteisto integroitiin elektroniikka-alustalle. Integroiminen toteutettiin myös ohjelmallisesti ohjelmointiympäristössä. Lopuksi tehtiin kartoitus rannetietokoneen virtalähderatkaisuista. Virtalähderatkaisu integroitiin seuraavaan prototyyppiin.

Rannetietokoneen laitteisto saatiin integroitua niin laitteisto- kuin ohjelmatasolla. Toistomäärälaskuri ja toistojen etsintä saatiin toteutettua kiihtyvyyssanturiteknologiaa hyväksikäyttäen. Lisäksi onnistuttiin vertailemaan eri virtalähderatkaisuja.

Asiasanat: SpoCo, rannetietokone, Arduino Atmega1280, elektroniikka-alusta

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Information Technology, Electronics Design and Testing

Author: Joel Syrjämäki

Title of thesis: The implementation of SpoCo's wrist computer second generation

Supervisor: Ensio Sieppi

Term and year when the thesis was submitted: Spring 2012

Pages: 40 + 5 appendices

The aim was to design and develop a wrist computer hardware and software for SpoCo's project in accordance with the requirements. Physical therapy professionals were also involved in the development of SpoCo's wrist computer, and people from the University of Lapland and the University of Jyväskylä. The wrist computer was designed to measure the movements at the gym, and then tell repeat count to the user. The goal of this project was to develop a prototype wrist computer to serve gym trainees and then finally market it.

The wrist computer project used the Arduino ATmega1280 development platform. Other wrist computer hardware was integrated into the electronic platform. The integration was carried out on software programming environment. The final step of the project was to compare the different power supply solutions to the wrist computer. Power supply solution was integrated into the next prototype.

The wrist computer hardware was integrated as in hardware than software levels. The count number of reps and the movement of the search was achieved with accelerometer sensor technology. In addition, we managed to compare the different power supply solutions.

Keywords: SpoCo, wrist computer, Arduino Atmega1280, electronic platform

ALKULAUSE

Opinnäytetyö tehtiin Oulun seudun ammattikorkeakoulun tekniikan yksikössä yhdessä SpoCo-tiimin kanssa. Opinnäytetyö on osa SpoCo-rannetietokoneen laitteiston ja ohjelmiston suunnittelu- sekä kehitysprojektia.

Kiitokseni haluan esittää SpoCo-rannetietokoneen toisen sukupolven toteutuksessa mukana olleille henkilöille, joiden kanssa tiimityöskentely on ollut opettavaista ja kehittävää. Erityiskiitokset yliopettaja Ensio Siepille opinnäytetyöni ohjauksesta ja Tuula Hopeavuorelle tekstinohjauksesta. Lisäksi haluan kiittää tutkijayliopettaja Manne Hannulaa, ja projektikoordinaattori Pekka Pohjosahoa työn tilauksesta. Opinnäytetyö auttoi kehittämään monipuolisesti tietoteknistä osaamistani. Projektin ansiosta opin paljon uutta laitteistosuunnittelusta ja ohjelmistokehityksestä.

Oulussa 31.5.2012

Joel Syrjämäki

SISÄLLYS

TIIVISTELMÄ	3
ALKULAUSE	5
1 JOHDANTO	7
2 ARDUINO MEGA-1280	8
3 SPOCO-RANNETIETOKONEEN LAITTEISTO	10
3.1 MPU-6000-mikropiiri	10
3.2 MPU-6000 ja FM25V05-mikropiirien integrointi piirilevyllä	11
3.3 BlueSMiRF-Bluetooth-yksikön kytkeminen Arduinoon	12
3.4 BlueSMiRF-Bluetooth-yksikön yhdistäminen tietokoneeseen	13
3.5 µOLED-96-G1-näyttömoduuli	17
3.6 Ramtrom FM25V05-FRAM lisämuisti	21
4 SPOCO-RANNETIETOKONEEN OHJELMISTO	23
4.1 Alkuasetukset ja taskien luominen	24
4.2 MPU-6000 antureiden lukeminen	24
4.3 Toistojen etsintä	26
4.4 Bluetooth-yhteys	26
4.5 Tiedontallentaminen muistiin	28
4.6 Spoco-rannetietokoneen ohjelmiston lataaminen Arduinoon	30
5 VIRTALÄHTEIDEN VERTAILU	31
5.1 Laitteiston virrankulutus	31
5.2 Valittu virtalähteratkaisu	35
5.3 Virtalähteen edut ja haitat	36
6 YHTEENVETO	38
LÄHTEET	40
LIITTEET	42

1 JOHDANTO

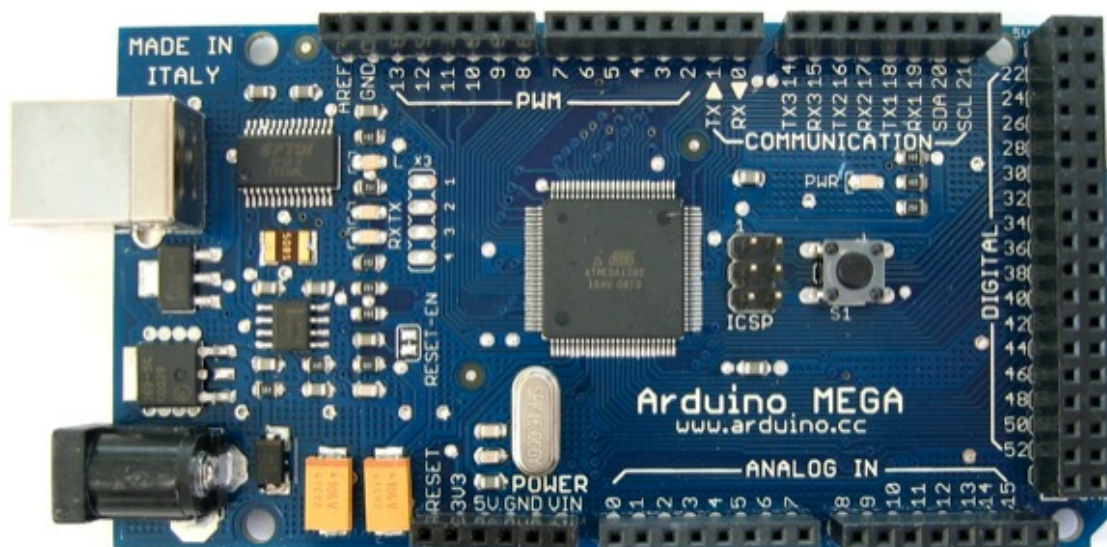
Opinnäytetyön tavoitteena oli suunnitella ja kehittää laitteistoa sekä ohjelmistoa työtilaajien vaatimusten mukaisesti. Projektin tavoitteena oli kehittää ohjelmistoa ja laitteistoa yhteensopivaksi. Laitteen päätarkoituksena oli mitata toistomääriä käyttäjän tehdessä erilaisia liikkeitä kuntosalilla. Toistomäärälaskuri perustuu MPU-6000-mikropiirin kiihtyvyysanturin ja gyroskoopin teknologiaan. Laitteisto- ja ohjelmistosuunnittelun oli tarkoitus mahdollistaa toimiva kokonaisuus seuraavaa generaatiota varten.

Laitteistosuunnittelussa perehdyttiin aluksi Arduino MEGA-1280 -elektroniikka-alustaan, jolla varsinainen projektin kehitys tapahtui. Ohjelmistoa kehitettiin Arduino-IDE-ympäristössä. Tarvittava kiihtyvyysanturi ja gyroskooppi sekä lisämuisti liitettiin elektroniikka-alustaan omalla lisäkortilla. Bluetooth-yksikkö ja μ OLED-näyttö liitettiin lisäksi myös elektroniikka-alustaan lisäkorteilla.

Keskeisimmässä osassa laitteistokehityksessä oli saada MPU-6000-mikropiiri, lisämuisti, μ OLED-näyttö ja Bluetooth-yksikkö integroitua yhtenäiseksi kokonaisuudeksi. Integrointi toteutettiin yhtenäiseksi niin laitetasolla kuin myös ohjelmatasolla. Projektin osien integrointi on osa tulevaa tuotteen jatkokehitystä, jossa perehdytään syvällisemmin siihen, millainen varsinaisen seuraavan generaation tuotteen tulisi olla.

2 ARDUINO MEGA-1280

Projektin kehitysalustana toimii kuvan 1 kaltainen Arduino MEGA-1280-elektroniikka-alusta. Alusta soveltui melko hyvin kehitettävän projektin tekemiseen, koska alustasta löytyi runsaasti ja monipuolisesti liitännöitä sekä ominaisuuksia. Erityisen hyvänä puolena oli se, että kehitettävän alustan IDE-ohjelmointiympäristö oli maksuton. Lisäksi IDE-ohjelmointiympäristöön löytyi runsaasti manuaaleja ja dokumentteja, sekä valmiita kirjastoja. Arduino MEGA-1280-elektroniikka-alusta oli hyvä valinta tähän projektiin, koska se on edullinen ja siitä löytyi kaikki tarvittavat ominaisuudet. IDE-ohjelmointiympäristö oli myös hyvä valinta projektiin, koska se oli käytettävyydeltään yksinkertainen ja sitä voitiin ohjelmoida USB-väylällä.



KUVA 1. MEGA-1280 elektroniikka-alusta (1, s. 1)

Arduino MEGA-1280 -elektroniikka-alustassa oli runsaasti ominaisuuksia, joita tarvittiin etenkin tässä projektissa. Alustalta löytyi Atmega1280-mikrokontrolleri, joka toimii 16 MHz:n taajuudella. Mikrokontrollerissa käytettiin 128 kbit:n flash-muistia ohjelmoitavalle koodille ja 4 kbit:ä bootloader-osiolle sekä 4 kbit:n muisti EEPROM-muistille. Alustasta löytyi runsaasti liitännöitä kuten 54 digitaalista

sisääntuloa, joista neljätoista toimi tarvittaessa pulssinleveysmoduloituina ulostuloina. Lisäksi alustalla oli 16 analogista sisääntuloa ja UART (Universal Asynchronous

Receiver Transmitter), jolla voitiin siirtää rinnakkaismuotoista tietoa sarjamuotoon ja päinvastoin. Lisäksi kortilta löytyi USB-väylä, jolla alusta liitettiin tietokoneeseen ohjelmointia varten, sekä ICSP-valmius ja virtaliitäntä. (1, s. 1.)

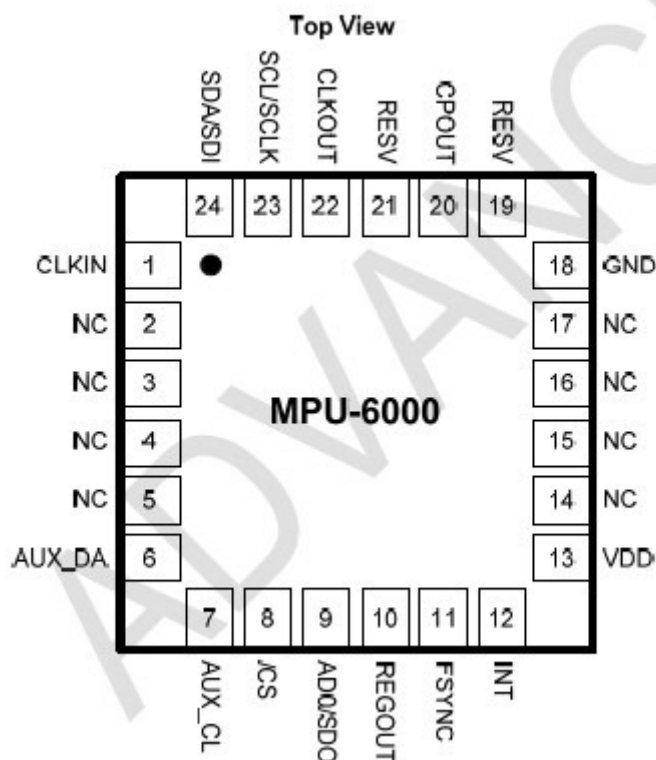
Tarkemmin ottaen alustalta löytyi myös kolme sarjaväylää TTL-sarjaliikenteelle sekä SPI- ja I²C-väylä. Lisäksi alustalla oli 8-bittinen pulssinleveysmodulaatio ulostulopinneissä 2-13 ja 44-46. Alustan käyttöjännite on 5 V:a ja I/O-pinnien maksimivirta 40 mA. Alustalta löytyi myös reset-painike ja automaattinen sulake, joka laukaistiin virran ylittäessä 500 mA:n rajan. (1, s. 1.)

3 SPOCO-RANNETIETOKONEEN LAITTEISTO

Projektissa käytettiin erilaisia oheislaitteita. Oheislaitteet integroitiin Arduino MEGA1280-elektroniikka-alustalle. Aluksi MPU-6000 ja Ramtrom FM25V05-FRAM -lisämuisti integroitiin alustaan, jonka jälkeen tehtiin Bluetooth-yksikön integrointi. Lopuksi integroitiin µOLED-näyttö ja FM25V05-FRAM -lisämuisti.

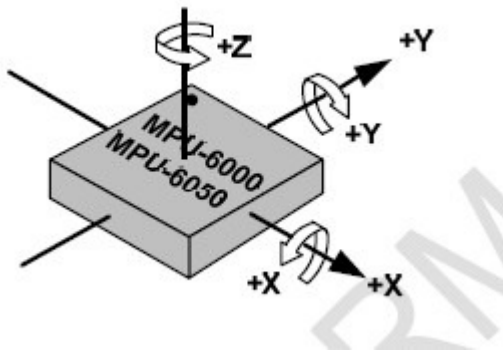
3.1 MPU-6000-mikropiiri

Projektissa käytettiin MPU-6000-mikropiiriä, jolla toteutettiin toistontunnistus ja toistontunnistuksen avulla kehitettiin toistomäärälaskuri. Mikropiirin pinnijärjestys näkyy kuvassa 2. Piiristä löytyy DMP-prosessori (digital motion processor), ja piiri käyttää tiedon välityksessä I2C-väylää. Piirin gyroskooppi kykeni tunnistamaan 250–2000 asteen liikkeen sekunnissa ja kiihtyvyysanturi tunnistasi $\pm 2g$ – $\pm 16g$ kiihtyvyydet.



KUVA 2. MPU-6000-mikropiirin pinnijärjestys (2, s. 21)

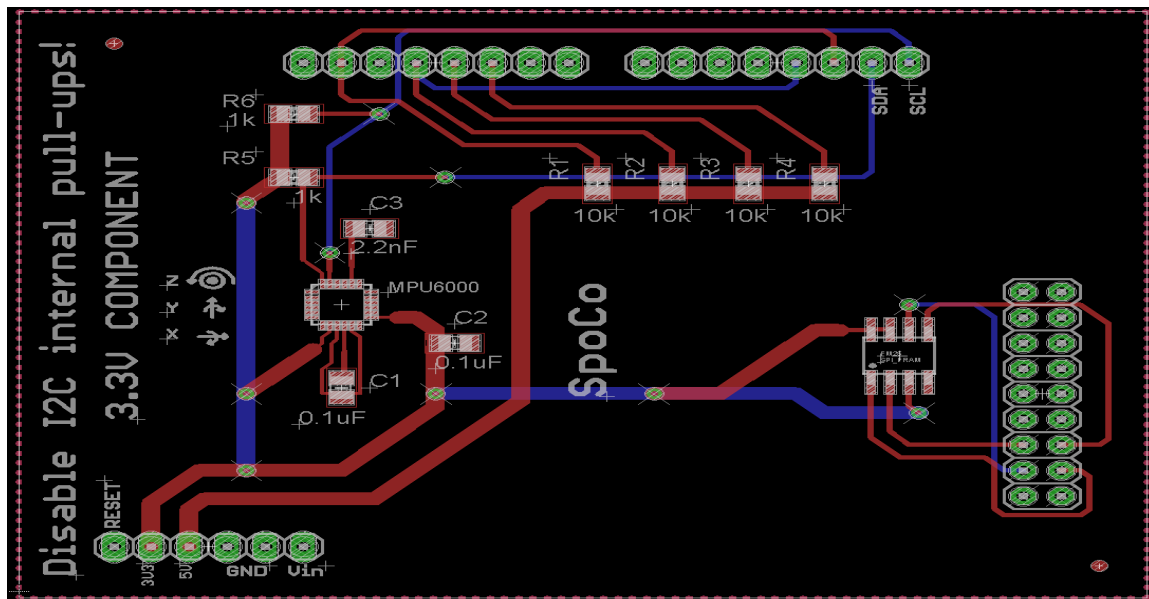
Kuvassa 3 näkyy akselien orientaatiot X, Y ja Z ja polariteetin kierto. Kuvassa 3 merkki (•) kuvaa mikropiirin pinnan 1 sijaintia. MPU-6000-mikropiirissä on yhdistetty 3-akselinen gyroskooppi ja 3-akselinen kiihtyvyysanturi samalle piille. Piiltä löytyy myös sisäinen lämpötila-anturi, jolla voidaan seurata piirin sisäistä lämpötilaa, ja tehdä lämpötilakalibrointi.



KUVA 3. MPU-6000 X, Y ja Z-akselien orientaatio ja polariteetin kierto (2, s. 21)

3.2 MPU-6000 ja FM25V05-mikropiirien integrointi piirilevylle

Ensimmäiseksi suunniteltiin itsesuunniteltu piirilevy, jonka layout näkyy kuvassa 4. Seuraavaksi hankittiin tarvittavat komponentit ja piirilevylle juotettiin MPU-6000-mikropiiri ja Ramtron FM25V05 FRAM -lisämuisti sekä tarvittavat vastukset, kondensaattorit ja piikkirimat. Piikkirimojen paikat suunniteltiin siten, että piirilevy voitiin liittää suoraan elektroniikka-alustalle kiinni. Piirilevyn yläpuolelle liitettiin myöhemmässä vaiheessa μ OLED-näyttö ja Bluetooth-yksikkö.



KUVA 4. Itsesuunniteltu piirilevy MPU-6000 ja FM25V05-mikropiireille

MPU-6000-mikropiiri toimii 2,3-3,4 V:n käyttöjännitteellä. Kuvassa 4 MPU-6000- ja FM25V05-mikropiiri kytkettiin kuvan 4 mukaisesti 3,3 V:n jännitteeseen. MPU-6000-mikropiiri ja sen ulkoiset komponentit kytkettiin piirilevyllä mikropiiriin manuaalista löytyvän kytkennän mukaisesti. MPU-6000-mikropiiriin ja elektronikka-alustan välisessä tiedonsiirrossa käytettiin I²C-väylää. MPU-6000-mikropiiriin SDA- ja SCL-pinnit kytkettiin piikkirimojen välityksellä elektronikka-alustan SDA- ja SCL-pinneihin, jotta I²C-väylä saatiin käyttöön.

3.3 BlueSMiRF-Bluetooth-yksikön kytkeminen Arduinoon

Projektissa käytettiin BlueSMiRF-Bluetooth-yksikköä, jonka oli kehittänyt SparkFun Electronics. BlueSMiRF toimii sarjaväylän RX- ja TX-linjojen kautta. Sarjaväylä toimii 2400-115200 bps nopeudella ja kantomatoksi luvataan avoimessa tilassa 106 m.

BlueSMiRF toimii 3,3-6,0 V:n käyttöjännitteellä. Pinnit yhdistetään Arduinoon ja BlueSMiRF kanssa kuvan 5 mukaisesti. Pinnien yhdistämisen jälkeen kytkettiin Arduinoon virrat päälle, jolloin BlueSMiRF-yksikön punainen LED alkaa vilkkua. Punainen LED ilmoittaa, että yksikkö oli kytketty päälle. Yksikön kytkemisen jälkeen se yhdistettiin tietokoneen Bluetooth-laitteen kanssa. (3, s. 3.)

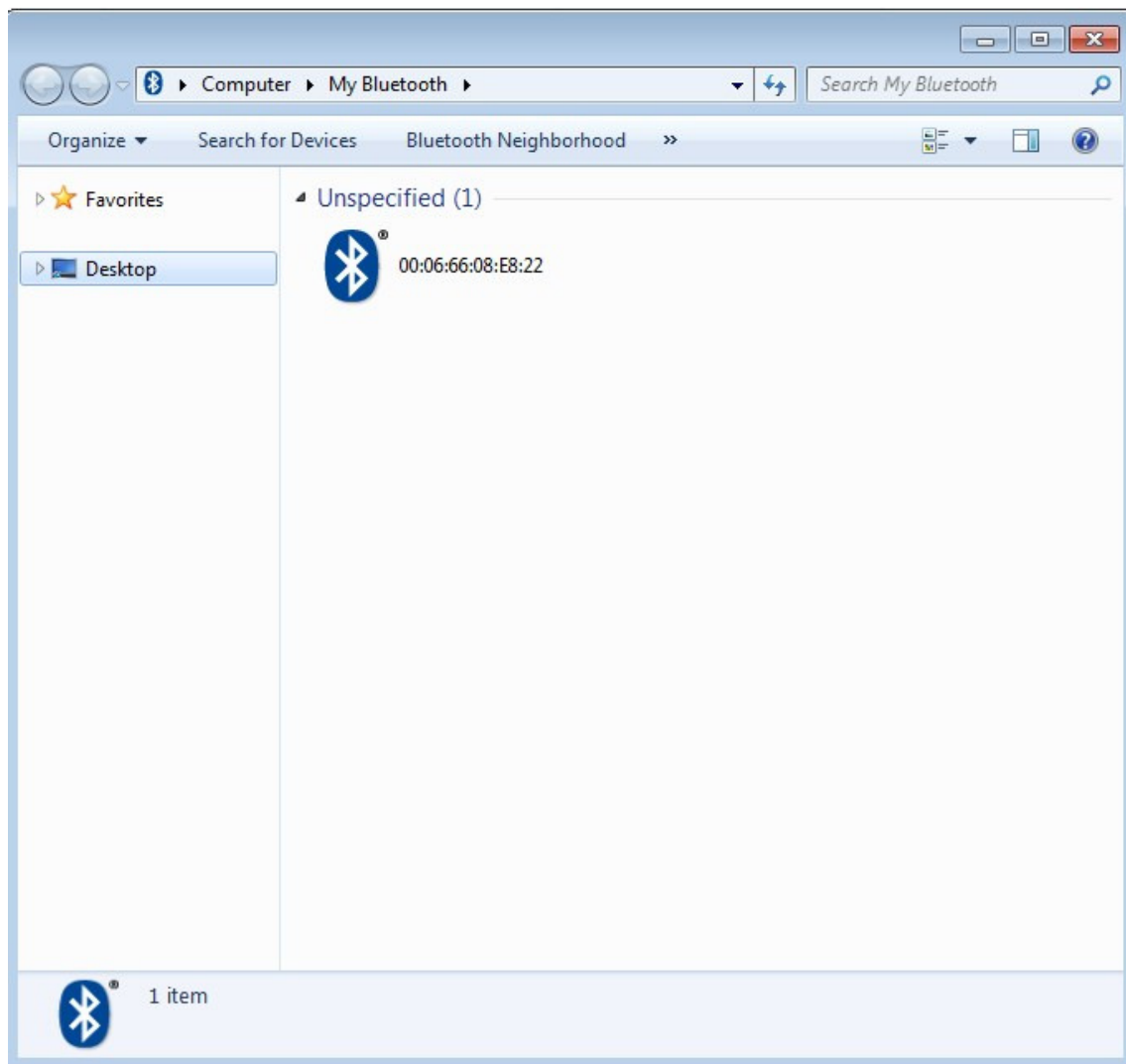
BlueSMiRF	Arduino Mega
VCC	5 V
GND	GND
RX	TX (or TX1, TX2, TX3)
TX	RX (or RX1, RX2, RX3)

Note: Always use the same pair of pins (e.g. TX1 and RX1, not TX1 and RX2)

KUVA 5. Pinnien yhdistäminen Arduinon ja BlueSMiRF-Bluetooth-yksikön välillä (3, s. 3)

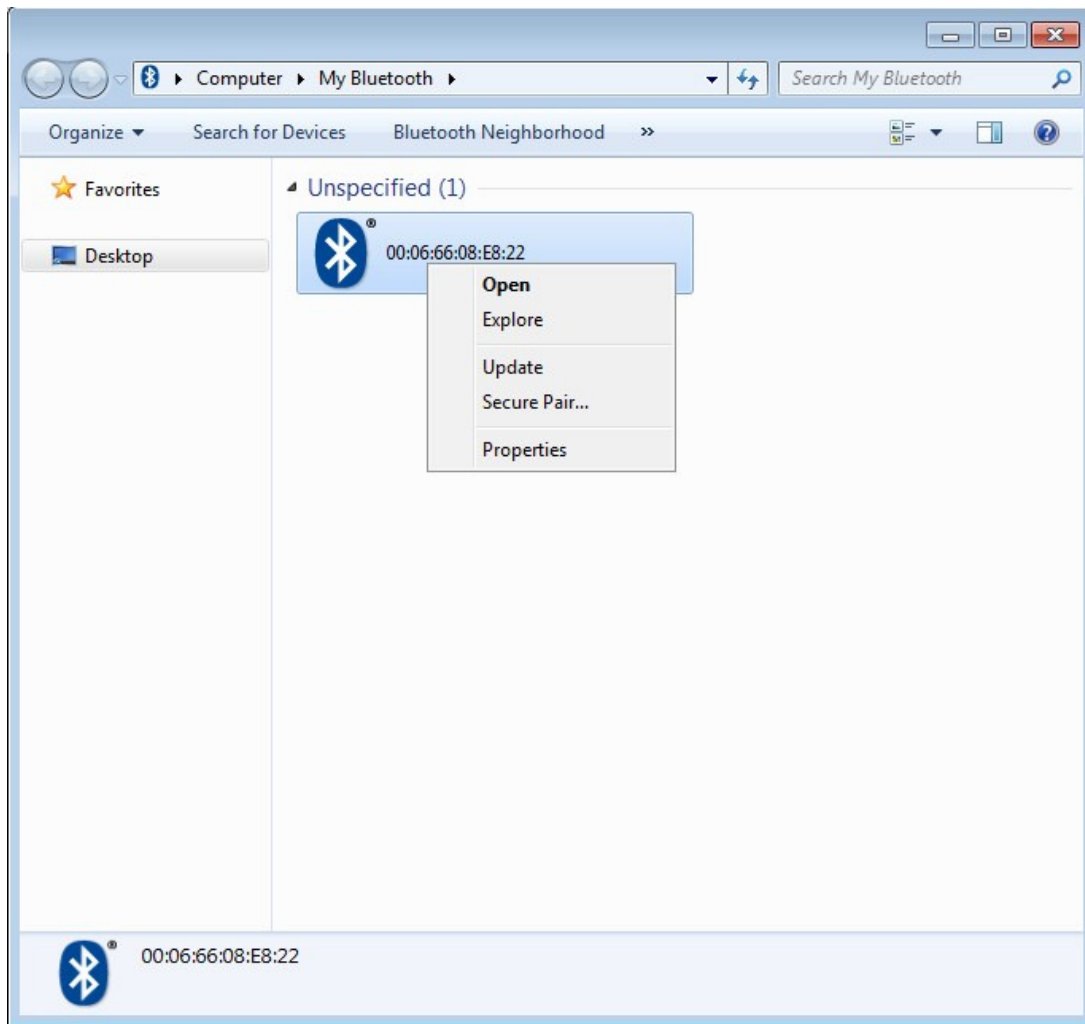
3.4 BlueSMiRF-Bluetooth-yksikön yhdistäminen tietokoneeseen

Aluksi tietokoneella skannattiin kantaman sisällä olevat Bluetooth-laitteet. Löydetyt Bluetooth-laitteet löytyivät "Computer → My Bluetooth" valikon alta. Bluetooth-laitteiden skannauksessa pitäisi löytyä kuvan 6 mukainen Bluetooth-laite.



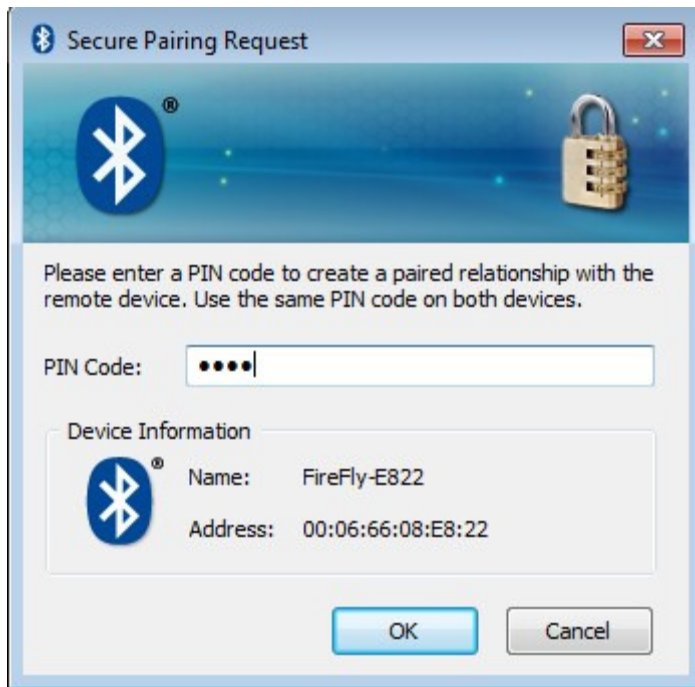
KUVA 6. My Bluetooth-valikko

Ennen kuin Bluetoothia voitiin käyttää tietokoneen kanssa, se paritettiin tietokoneen Bluetooth-laitteeseen. Paritus onnistui yksinkertaisesti siten, että Bluetooth-valikosta haettiin kuvan 7 mukaisesti "Secure pair"-toiminto.



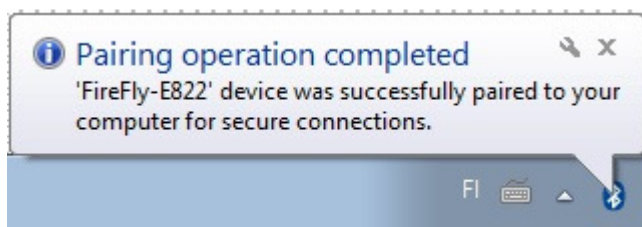
KUVA 7. My Bluetooth-valikon "Secure pair.."-toiminto

Tämän jälkeen PIN-koodia kysyvä dialogi ilmestyi näytölle kuvan 8 mukaisesti. PIN-koodiksi syötettiin yksikön alkuperäinen PIN-koodi eli "1234".



KUVA 8. PIN-koodin syöttäminen

Koodin syöttämisen jälkeen BlueSMiRF-Bluetooth-yksikön piti ilmoittaa "Pairing operation completed". ja ilmoitus siitä ilmestyi työpöydän alareunaan kuvan 9 mukaisesti. Nyt Bluetooth-yksikkö voitiin ottaa käyttöön Termite-ohjelmassa.



KUVA 9. Paritusoperaatio on onnistunut

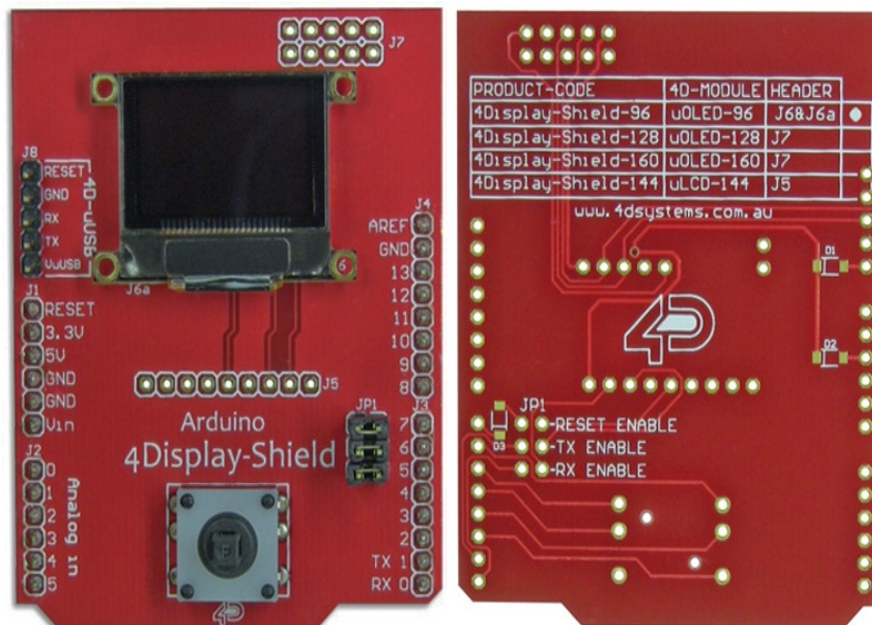
Mikäli parituksessa epäonnistuttiin, käyttöjärjestelmä antoi kuvan 10 mukaisen virheilmoituksen. Virheilmoitus johtui todennäköisesti silloin väärin syötetystä PIN-koodista. Virheilmoituksen tultua tarkastettiin myös mahdollinen virheellinen kytkentä.



KUVA 10. Epäonnistunut paritusoperaatio

3.5 μ OLED-96-G1-näyttömoduuli

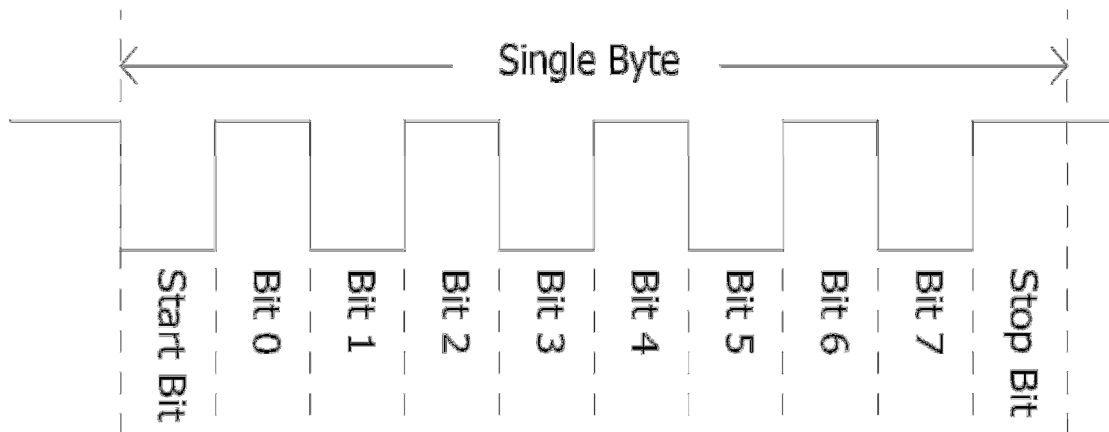
Projektissamme käytettiin kuvan 11 mukaista 4D Systems μ OLED-96-G1-näyttömoduulia. Moduuli käyttää OLED-teknologiaa ja näyttömoduulissa on GOLDELOX-SGC-sarjagrafiikkakontrolleri. Moduuli tarjosi paljon itsenäisiä toimintoja erilaisille projekteille. Moduulia käytettiin projektissamme, koska moduuli oli helppo integroida käyttäen sarjaväylää elektroniikka-alustan mikrokontrollerin ja näyttömoduulin välillä. (4, s. 2.)



KUVA 11. 4D Systems μ OLED-96-G1-näyttömoduuli (5)

μ OLED-96-G1-näyttömoduuli perustui laitetasolla UART (Universal Asynchronous Receiver Transmitter)-sarjaliikennepiiriin. Sarjaliikennepiirin avulla näyttömoduuli ja mikrokontrolleri kommunikoivat keskenään. Tieto

lähetettiin 8-bittisenä kuvassa 12 esitetyllä tavalla. Lähetykseen käytettiin TX-pinniä ja vastaanottoon RX-pinniä. Alussa lähetettiin aloitusbitti ja lopussa lopetusbitti. Bittinopeus moduulissa oli 300-256000 bit/s. (6, s. 5.)



KUVA 12. Lähetettävä tavu bitteinä (6, s. 5)

Näyttömoduulin pinnit ja niiden selitykset näkyvät kuvassa 13. Asynkronisen sarjaväylän TX-pinni eli numero 2 kytkettiin elektroniikka-alustan RX-pinniin. Elektroniikka-alustalle välitettiin tietoa tämän pinnin kautta näyttömoduulista. Sarjaväylän RX-pinni eli numero 3 kytkettiin elektroniikka-alustan TX-pinniin. Kyseisen pinnin kautta elektroniikka-alustalta välitettiin tietoa näyttömoduuliin. VCC eli pinni numero 1 kytkettiin elektroniikka-alustan 5,0 V:n jännitteeseen. GDN-pinnit eli numerot 4 ja 6 kytkettiin elektroniikka-alustan maapotentiaaliin. RESET-pinni eli numero 5 toimi silloin, kun aktiivinen pulssi oli alhaalla enemmän kuin 2 mikrosekuntia. SWITCH-pinni eli numero 7 jätettiin tässä projektissa käyttämättä. (6, s. 4.)

Pin	Symbol	I/O	Description
1	VCC	I	Main Voltage Supply +ve input pin. Reverse polarity protected. Range is 4.0V to 5.5V, nominal 5.0V.
2	TX	O	Asynchronous Serial Transmit pin. Connect this pin to host micro-controller Serial Receive (Rx) signal. The host receives data from uOLED-96-G1(SGC) via this pin. This pin is tolerant up to 5.0V levels.
3	RX	I	Asynchronous Serial Receive pin. Connect this pin to host micro-controller Serial Transmit (Tx) signal. The host transmits commands and data to the uOLED-96-G1(SGC) via this pin. This pin is tolerant up to 5.0V levels.
4	GND	P	Supply Ground.
5	RESET	I	Master Reset signal. Internally pulled up to 3.3V via a 4.7K resistor. An active Low pulse greater than 2 micro-seconds will reset the module. If the module needs to be reset externally, only use open collector type circuits. This pin is not driven low by any internal conditions. The host should control this pin via one of its port pins using an open collector/drain arrangement.
6	GND	--	Ground. Also provides an easy ground connection of pin 7 using a 0.1" shunt to auto-run a slide show script program on power-up.
7	SWITCH	I	Multi Button or Joystick switch input pin. Option is available to connect from 1 up to 5 push buttons. If connected to GND on power-up it will auto-run a script program from the memory card.

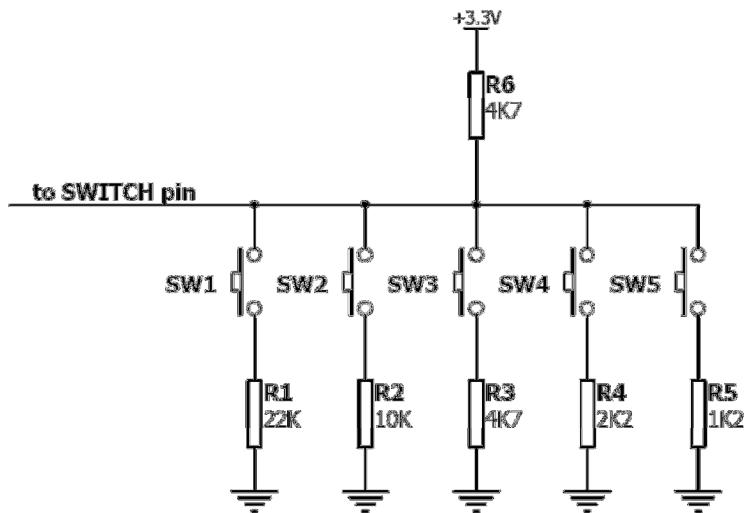
KUVA 13. Näyttömoduulin pinnit (6, s. 4)

Button-Switch-käyttöliittymää voitiin käyttää näyttömoduulissa käyttämällä viittä eri painonappia tai viittä eri tilaa multi-switch-toiminnolla. Tämä tapahtui siten, että jokaista painonappia kohti oli omat vastusarvot, jotka näkyvät kuvassa 14. (6, s. 5.)

Number of Buttons	Button Number	Resistor Value
1	SW1	22K
2	SW2	10K
3	SW3	4.7K
4	SW4	2.2K
5	SW5	1.2K

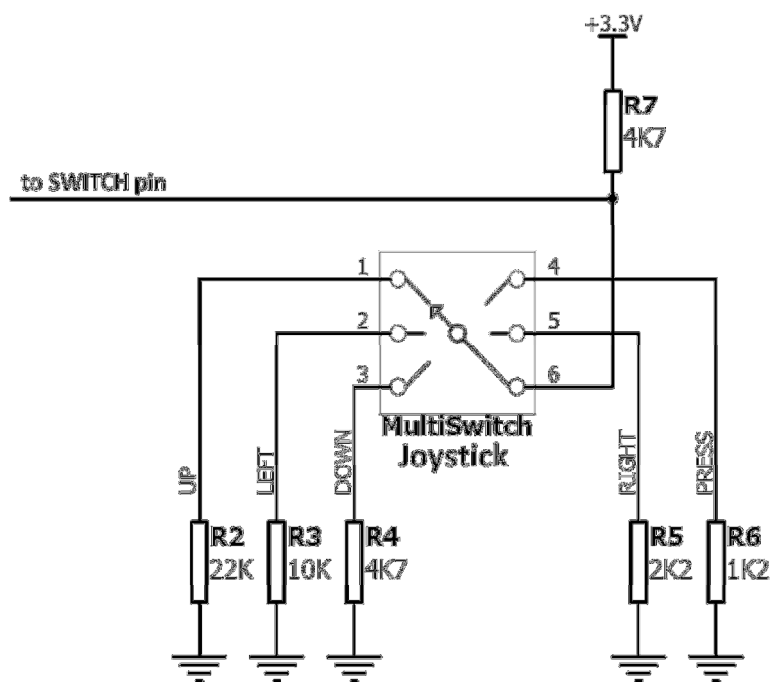
KUVA 14. Painonappien vastusarvot (6, s. 5)

Kytkenäkaavio painonapeille näkyy kuvassa 15. Painonapit kytkettiin vastuksien kautta maapotentiaaliin.



KUVA 15. Painonappien kytkentäkaavio (6, s. 5)

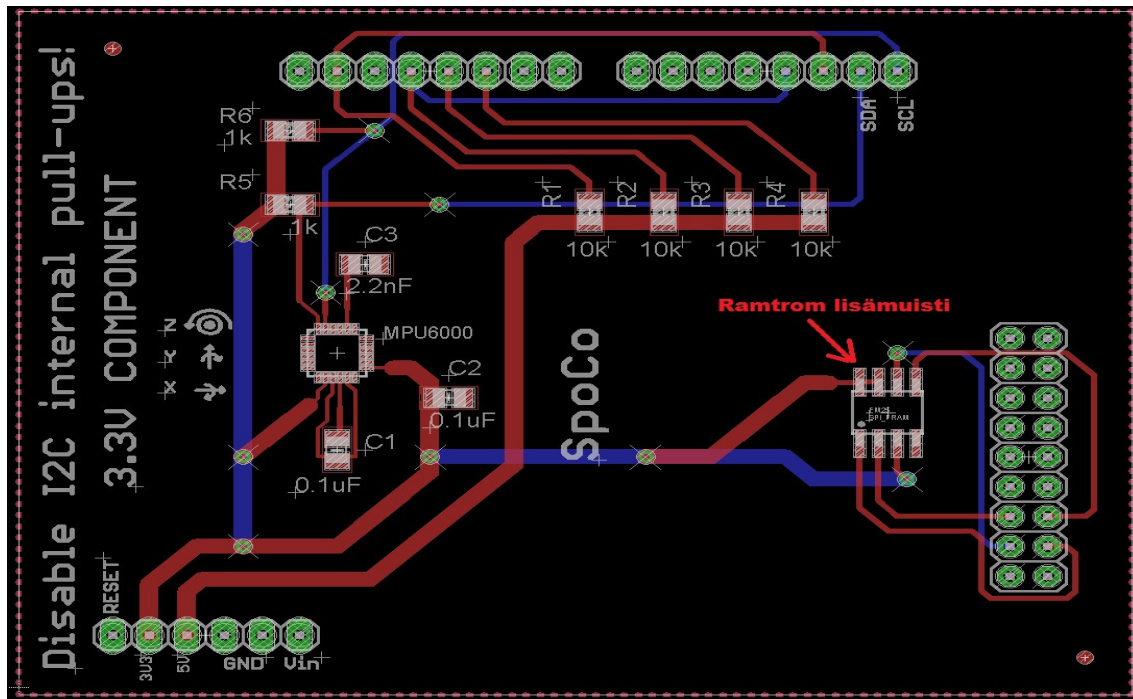
Vastaavasti MultiSwitch-toiminnolle kytkentäkaavio näkyy kuvassa 16. Kytkentäkaaviossa MultiSwitch-pinnit kytkettiin vastuksien kautta maapotentiaaliin.



KUVA 16. MultiSwitch- kytkentäkaavio (6, s. 6.)

3.6 Ramtrom FM25V05-FRAM lisämuisti

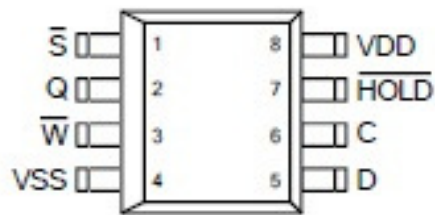
Projektiimme otettiin käyttöön kuvan 17 mukainen Ramtrom FM25V05 FRAM -lisämuisti tallennuskapasiteetin kasvattamiseksi. Muistiin voitiin tallentaa myöhemmässä vaiheessa tarvittavia harjoitustiedostoja. Lisämuisti sijoitettiin MPU-6000-mikropiiriin tavoin itsesuunnitellulle piirilevylle.



KUVA 17. Ramtrom-muisti piirilevyllä

Lisämuistissa oli 512 kilobittiä tallennuskapasiteettia ja kommunikaatio elektroniikka-alustan ja lisämuistin välillä toteutettiin SPI-väylän kautta. Lisämuistin kotelotyyppiä valittiin 8-pinninen SOIC (Small-outline Integrated Circuit), jonka pinnijärjestys sekä selitykset esitetään kuvassa 18. Piirivalinta tapahtui 1-pinnillä, jolla valittiin otettiinko lisämuisti käyttöön vai ei. 0-tilalla lisämuisti otettiin käyttöön ja 1-tilalla lisämuisti laitettiin valmiustilaan. Sarjatietaa välitettiin muistin lukutilassa ulos pinnistä 2. Muistiin kirjoittaminen estettiin pinnillä 3, joka tapahtui siten, että pinni 3 laitettiin 0-tilaan. Maapotentiaali liitettiin pinniin 4 ja käyttöjännite 3,3 V:aa pinniin 8. Lisämuisti pysäytettiin tarvittaessa syöttämällä 0-tila pinniin 7, mikäli prosessorin haluttiin tekevän

välillä muita taskeja. Kellotulo kytkettiin pinniin 6 ja tietoa kirjoitettiin muistiin pinnistä 6.



Pin Name	Function
/S	Chip Select
/W	Write Protect
/HOLD	Hold
C	Serial Clock
D	Serial Data Input
Q	Serial Data Output
VDD	Supply Voltage
VSS	Ground

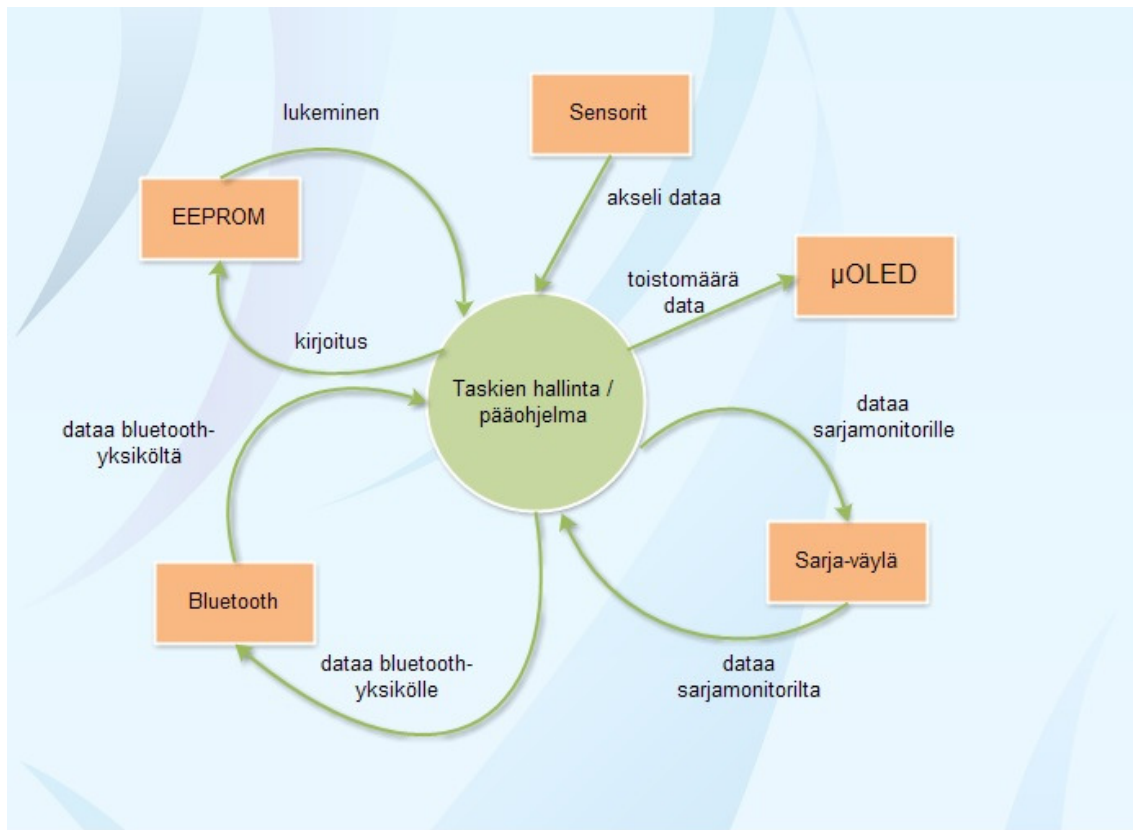
KUVA 18. Ramtron-muistin pinnijärjestys ja pinnien selitykset (6, s. 1)

4 SPOCO-RANNETIETOKONEEN OHJELMISTO

Pääohjelma toteutettiin siten, että toistojen etsinnälle, Bluetooth-yhteydelle, antureiden lukemiselle ja tiedontallentamiselle tehtiin omat taskit. Taskit priorisoitiin kukin tärkeysjärjestykseen. Toistomäärälaskurin testaus tapahtui sarjamonitorilla ja Bluetooth-yhteyden testaus tehtiin Termite-ohjelmalla, koska μ OLED-96-G1-näyttömoduulin integroinnissa ilmeni ongelmia.

Kuvaan 19 esitettiin rannetietokoneen ohjelmiston liittymäkaavio. Suorakaiteen muotoiset laatikot tehtiin kuvaamaan terminaattoreita, jotka kuvaavat niitä rajapinnan olioita, joista tulee järjestelmään tietoa ja, joille järjestelmä generoi dataa. Keskellä olevalla pallolla kuvattiin kehitettävää järjestelmää, eli rannetietokoneen pääohjelmaa. (11, s. 2.)

Kuvassa 19 kuvataan ohjelmistoon liittyvät terminaattorit. Terminaattorilla (Sensorit) kuvattiin, että sensorien akseleilta luettiin dataa silloin, kun sensorien lukutaski laitettiin aktiiviseksi. Terminaattorilla (EEPROM) kuvataan tiedon lukemista EEPROM-muistista ja kirjoittamista EEPROM-muistiin silloin, kun tiedontallentamisen taski laitettiin aktiiviseksi. Terminaattorilla (Bluetooth) kuvattiin tiedon siirtämistä elektroniikka-alustalta sarjaväylän avulla Bluetooth-yksikölle ja päinvastoin. Lisäksi terminaattorilla (Sarja-väylä) kuvattiin tiedon siirtämistä elektroniikka-alustalta sarjaväylään ja päinvastoin. Viimeisellä (μ OLED) terminaattorilla kuvattiin toistomäärä datan siirtämistä μ OLED-näytölle.



KUVA 19. Liittymäkaavio ohjelmistosta

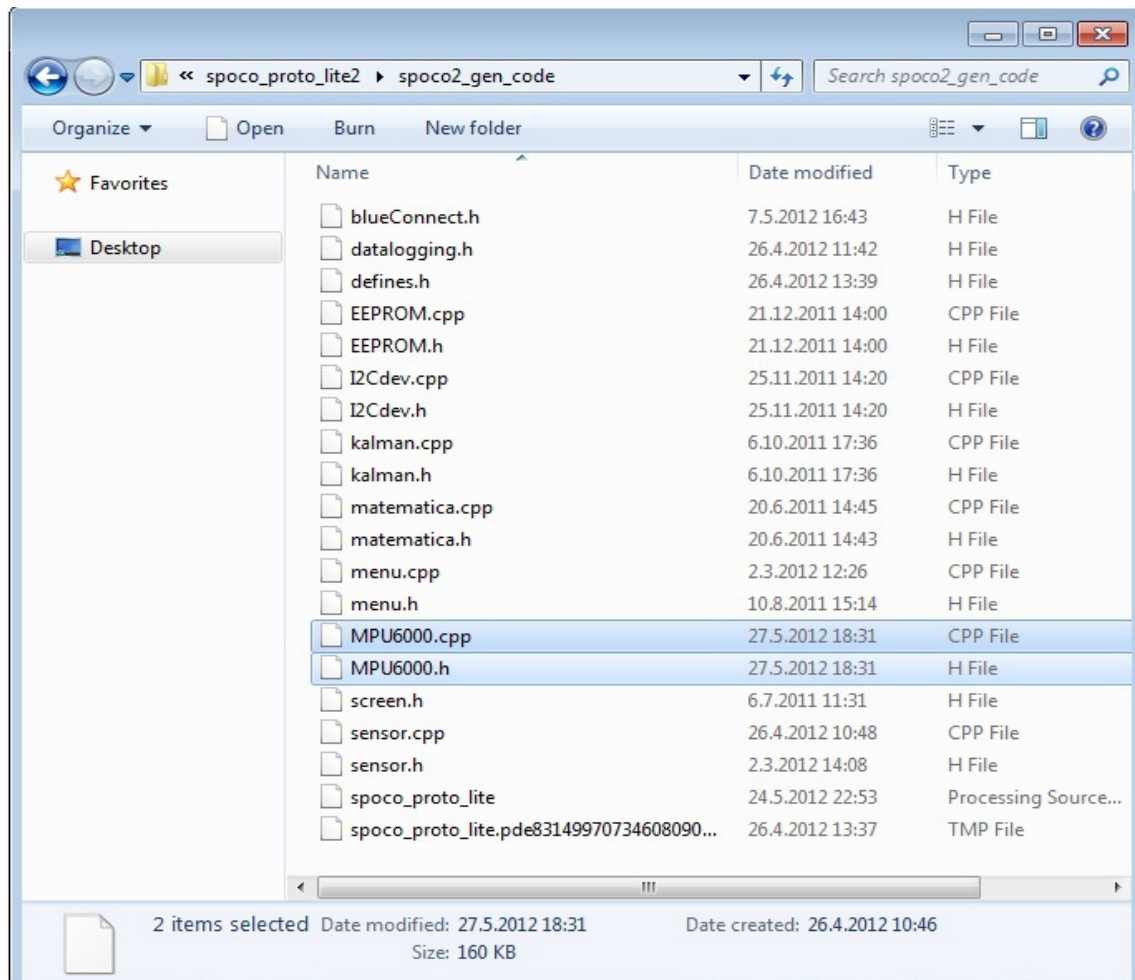
4.1 Alkuasetukset ja taskien luominen

Pääohjelmassa aluksi sisällytettiin pääohjelmaan tarvittavat kirjastot. Tämän jälkeen lisättiin globaalit muuttujat ja ilmoitettiin käytettävät taskit. Void `setup()` -kohdassa aluksi initialisoitiin MPU-6000-mikropiiri käyttöön. Alustamisen jälkeen otettiin käyttöön Serial- ja Serial1-sarjaväylät. Lopuksi luotiin tarvittavat taskit, joita olivat `(readSensor)`, `(findRepeat)`, `(dataLogging)` ja `(blueConnect)`. Jokaiselle taskille asetettiin käytettävät prioriteetit.

4.2 MPU-6000 antureiden lukeminen

Aluksi varsinaiseen ohjelmakansioon lisättiin MPU-6000-mikropiirin tarvitsema otsikkotiedosto `MPU6000.h` ja lähdetiedosto `MPU6000.cpp` kuten kuvasta 20 nähdään. Otsikkotiedostoon sisällytettiin kaikki tarvittavat määrittelyt MPU-6000-mikropiirille ja lähdetiedostoon sen sijaan sisällytettiin varsinaista MPU-

6000-mikropiirin koodia. Lisäksi sensor.h tiedostoon lisättiin rakenne (struct) joka luettiin anturilta.



KUVA 20. Lisätty otsikkotiedosto MPU6000.h ja lähdetiedosto MPU6000.cpp

Pääohjelmassa luotiin aluksi anturin lukutaski `declareTaskLoop(readSensor); TaskLoop(readSensor)`, eli anturinlukutaskissa aluksi herätettiin taski oikealla tajuudella ja initialisoitiin `xLastExecutionTime`, jotta `vTaskDelayUntil()` toimisi oikein. Sen jälkeen laitettiin kirjoituskohdat buffereissa nolaksi ja nollattiin myös anturilta saadut näytteet. Lopuksi ilmoitettiin, että taski on käynnissä ja määriteltiin 16-bittiset kokonaislukumuuttujat `ax`, `ay`, `az`, `gx`, `gy` ja `gz`.

While-silmukassa määriteltiin, että taskia ajettiin niin kauan kuin taskia tarvittiin, ja määriteltiin, että taskia ajetaan anturin taajuudella. Seuraavaksi haettiin

liikedataa akseleilta ja kirjoitettiin anturin dataa bufferiin. Testausta varten luotiin lopuksi muuttujat testaus[0]-testaus[5] sarjamonitoria varten, johon akseleiden liikedatat tulostettiin. Lopussa myös siirryttiin bufferissa eteenpäin ja hypättiin bufferin alkuun mikäli oltiin bufferin lopussa. Lisäksi indeksiä kasvatettiin yhdellä, ja lopulta tultiin ulos while-silmukasta sekä tuhottiin taski kuluttamasta muistia turhaan.

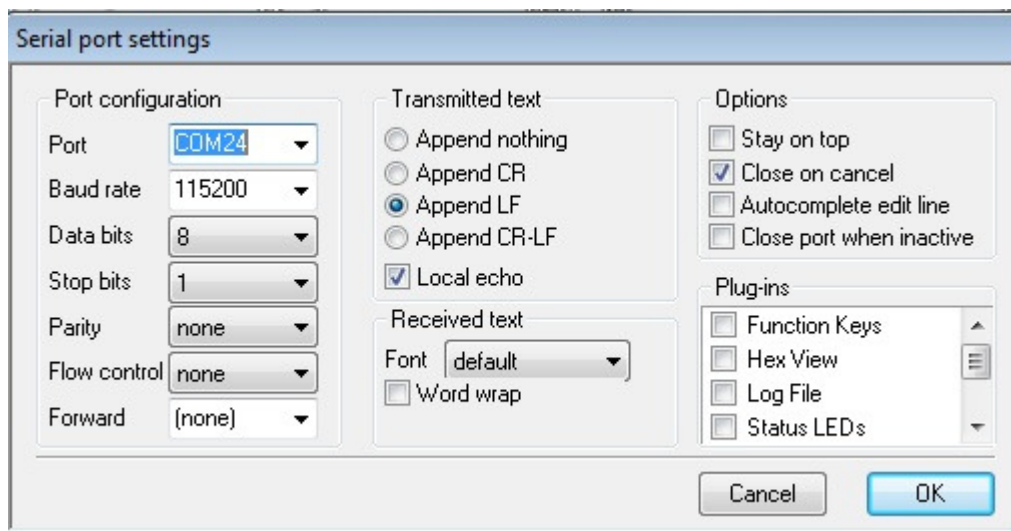
4.3 Toistojen etsintä

Toistonetsintä eli taskLoop(findRepeat) taskissa etsittiin toistoja kiihtyvyysanturin akseleita hyväksi käyttäen. Aluksi tehtiin tarvittavat määrittelyt ja asetettiin muuttujille tarvittavat alkuarvot. While-silmukassa asetettiin taski aktiiviseksi ja tehtiin tarvittavat binääritaulukot. Binääritaulukon arvoilla haettiin oikea arvo kiihtyvyyden poikkeamalle. Sen jälkeen etsittiin liikkeen aloituskohtaa. Mikäli data poikkesi nolasta, acc_bitin arvoksi asetettiin true ja lippu kerrottiin, etsittiinkö liikkeen aloitusta (false) vai liikkeen lopetusta (true). Liike aloitettiin, mikäli laskuri tavoitti LIIPASIN_ALOITUS arvon. Sitten lippu asetettiin arvoon true, koska sillä aloitettiin liikkeen lopetuskohdan etsintä. Mikäli liikkeen lopetus tapahtui, siirryttiin taas etsimään seuraavan liikkeen aloituskohtaa ja lippu asetettiin arvoon false, sekä laskuri nollattiin ja toistojen määrää kasvatettiin yhdellä.

4.4 Bluetooth-yhteys

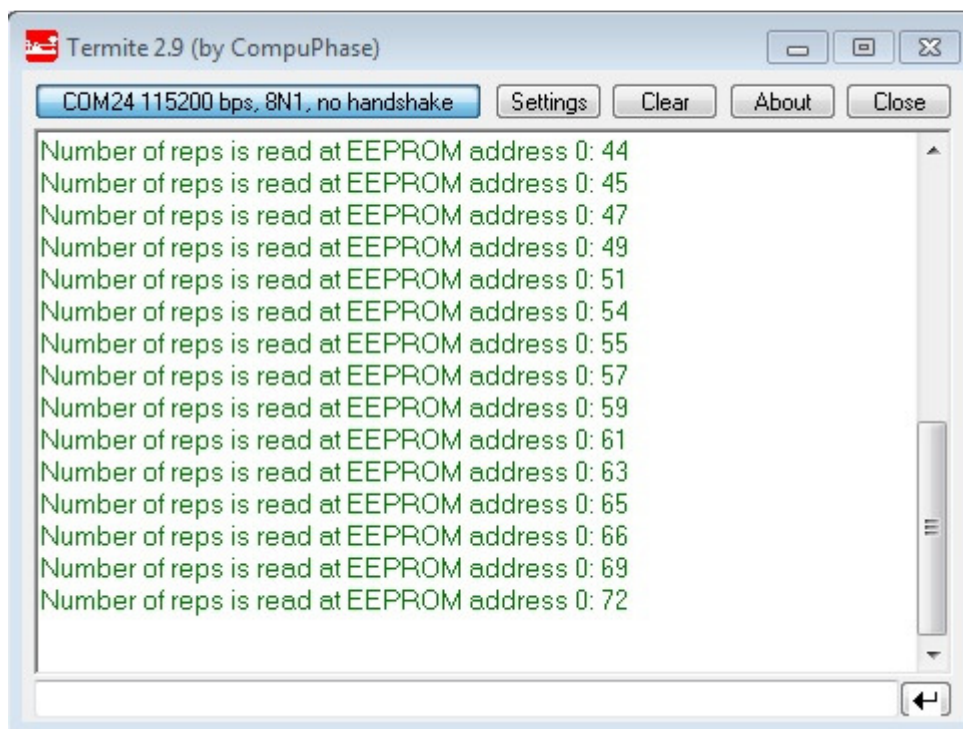
Bluetooth-yhteyden taskissa taskLoop(blueConnect) aluksi määriteltiin muuttujat blueByte Bluetooth-yhteydelle, ja usbByte; sarjaväylän yhteydelle. Seuraavana aloitettiin kuuntelemaan sarjaväylää Serial1, ja siitä voitiin lukea Bluetooth-yhteyden dataa tietokoneen ja Bluetooth-yksikön välillä.

Ohjelman lataamisen jälkeen Bluetooth-yksikön sarjaväylää tarkkailtiin Termite-ohjelmalla. Termite-ohjelmassa valittiin bluetooth-yksikön käyttämä sarjaportti kuvan 21 mukaisesti.



KUVA 21. Sarjaportin asettaminen

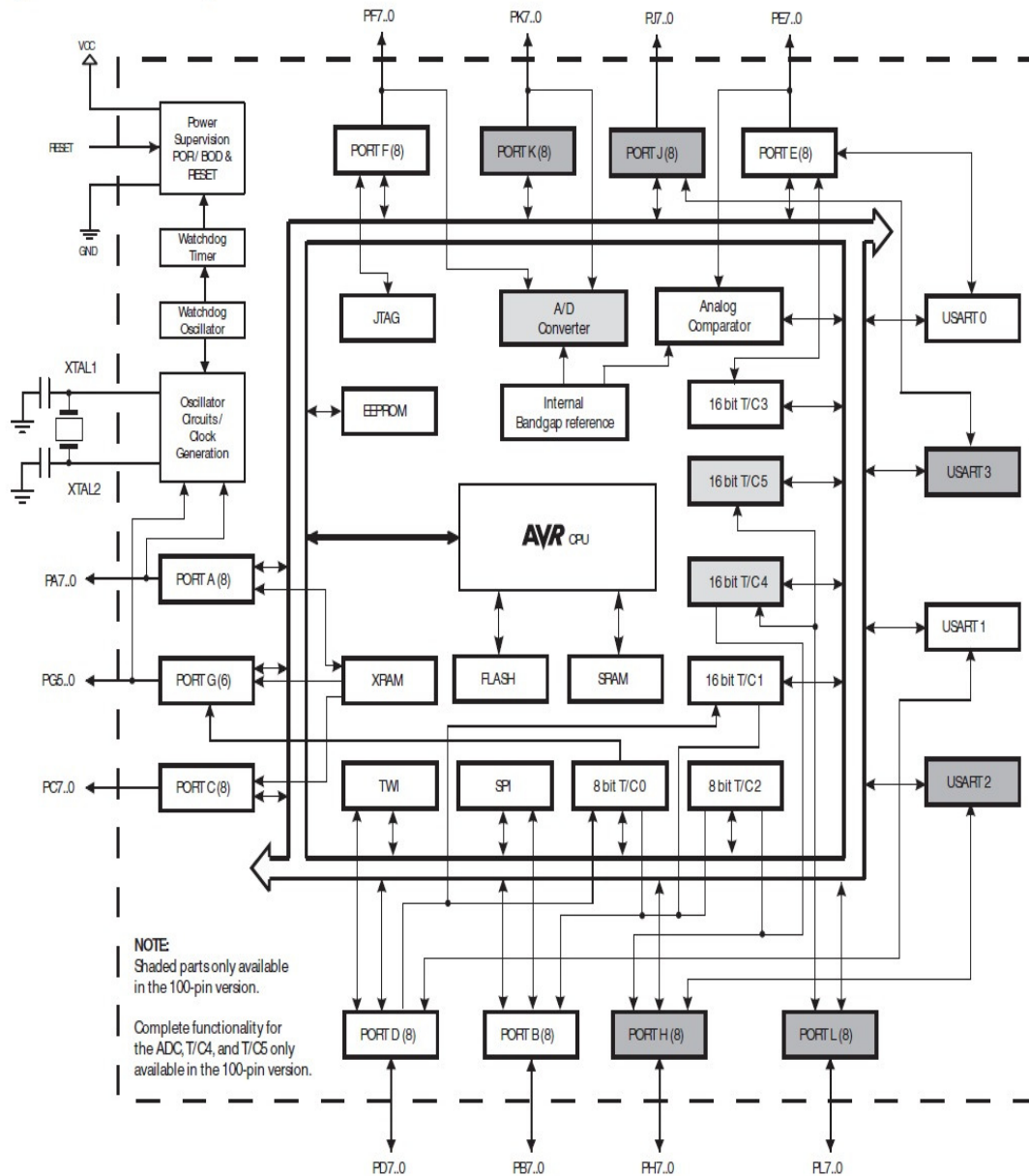
Sarjaportin asetuksen jälkeen Termite-ohjelma alkoi näyttämään toistojen määrää kuvan 22 mukaisesti. Tieto välittyi Bluetooth-yksikön ja tietokoneen välillä.



KUVA 22. Termite ohjelmaan tulostettiin Bluetooth-sarjaväylän dataa

4.5 Tiedontallentaminen muistiin

Tiedontallentaminen tässä projektissa toteutettiin siten, että toistomäärädata tallennettiin ainoastaan elektroniikka-alustan EEPROM-muistiin. Ajan puutteen johdosta Ramtron FM25V05 FRAM -lisämuistia ei päästy ottamaan projektiimme ohjelmiston puolelta mukaan. Kuvassa 23 esitettiin ATMEGA1280 mikrokontrollerin lohkokaavio. Lohkokaaviosta nähdään, että käytännössä EEPROM-muisti sijaitsee mikrokontrollerin sisällä. EEPROM-muistina käytettiin 4 kilobitin kokoista muistia, johon mahtui 512 tavua tietoa eli 4096 bittiä. EEPROM-muistiin kirjoitettiin tehtyjen toistojen määrä reaaliaikaisesti. Dataa luettiin EEPROM-muistista, josta se välitettiin sarjaväylään tai vaihtoehtoisesti Bluetooth-yhteyden sarjaväylään luettavaksi.



KUVA 23. ATmega1280 lohkokaavio (7, s. 5)

Tiedontallentaminen toteutettiin tiedontallennus taskilla, eli taskLoop(dataLogging) taskissa. Aluksi tehtiin tarvittava otsikko- ja lähdetiedosto, jotta tiedontallentaminen ja lukeminen onnistuisi EEPROM-muistista pääohjelmassa. Lisäksi otsikkotiedostossa määriteltiin haluttu kirjoitettavan tiedon koko kahdeksan tavun kokoiseksi. Aluksi ohjelmoitiin otsikkotiedostossa kahden tavun muuttujan tallentaminen EEPROM-muistiin.

Tämän jälkeen ohjelmoitiin vastaavasti kahden tavun muuttujan lukeminen EEPROM-muistista.

Pääohjelmassa `taskLoop(dataLogging)` toteutettiin toistojen määrän kirjoittaminen. Kirjoittaminen tapahtui siten, että toistojen määrä kirjoitettiin reaaliaikaisesti EEPROM-muistin kohtaan 0.

4.6 Spoco-rannetietokoneen ohjelmiston lataaminen Arduinoon

Pääohjelman lataaminen suoritettiin Arduino elektroniikka-alustalle siten, että Bluetooth-yksiköstä otettiin jännitteet aluksi pois. Jännitteet otettiin pois, koska silloin ohjelmiston lataaminen onnistui ilman virheilmoituksia. Lisäksi Termite-ohjelmisto suljettiin pois päältä, ettei sarjaväylien välillä tapahdu ristiriitoja. Ohjelmiston latauksen jälkeen avattiin Termite-ohjelma onnistuneesti. IDE-ohjelman sarjamonitorilla ei kuitenkaan voitu seurata Bluetooth-sarjaväylän tiedonkulkua, vaan pystyttiin seuraamaan ainoastaan IDE-ohjelman sarjaväylän tiedonkulkua.

5 VIRTALÄHTEIDEN VERTAILU

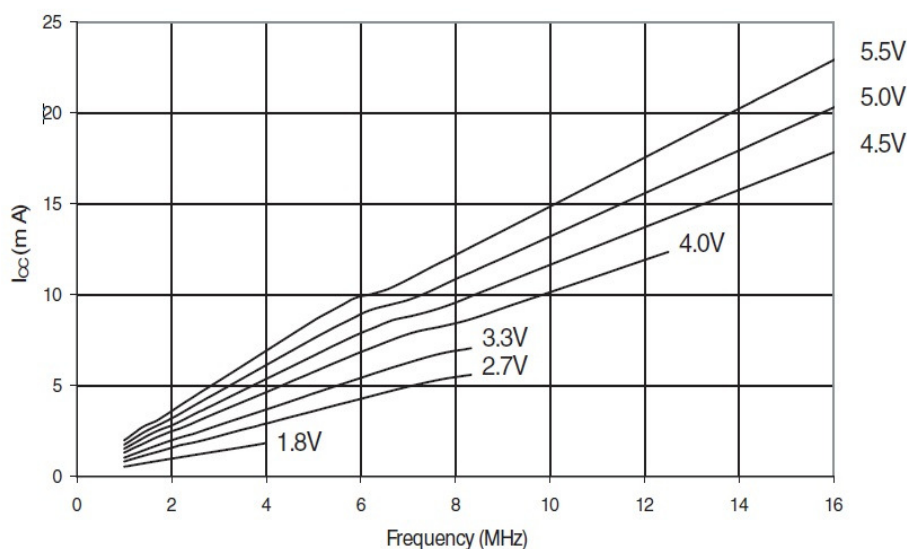
Projektiin löydettiin paljon ja monipuolisesti erilaisia virtalähderatkaisuja. Virtalähteeksi valittiin sellainen virtalähde, joka voitiin liittää käteen asennettavan rannetietokoneen yhteyteen. Lisäksi virtalähteen tulisi kestää käyttöä mielellään useita tunteja. Lisäksi virtalähteelle asetettiin kriteereiksi, että virtalähde ei saisi olla hinnaltaan liian kallis, ja virtalähteen piti olla uudelleen ladattava.

5.1 Laitteiston virrankulutus

Projektissa selvitettiin myös karkeasti laitteiston virrankulutusta, jonka perusteella saatiin laskettua tarvittava virtalähteen kapasiteetti. Atmega1280 virrankulutus aktiivisessa tilassa, ja 5,0 V:n käyttöjännitteellä, sekä 16 MHz:n kellotaajuudella oli kuvan 24 perusteella noin 21 mA.

ATmega640/1280/1281/2560/2561

Figure 32-2. Active Supply Current vs. Frequency (1MHz - 16MHz)



KUVA 24. ATmega1280 virrankulutus eri kellotaajuuksilla (7, s. 385)

Gyrokoopin ja kiihtyvyyssanturin virrankulutus, kun DMP-prosessointi oli kytketty pois päältä, oli kuvan 25 perusteella 3,8 mA.

PARAMETER	CONDITIONS	MIN	TYP	MAX	Units	Notes
TEMPERATURE SENSOR						
Range			-40 to +85		°C	
Sensitivity	Untrimmed		340		LSB/°C	
Temperature Offset	35°C		-521		LSB	
Linearity	Best fit straight line (-40°C to +85°C)		±1		°C	
VDD POWER SUPPLY						
Operating Voltages		2.375		3.46	V	
Normal Operating Current	Gyroscope + Accelerometer + DMP		3.9		mA	
	Gyroscope + Accelerometer (DMP disabled)		3.8		mA	
	Gyroscope + DMP (Accelerometer disabled)		3.7		mA	
	Gyroscope only (DMP & Accelerometer disabled)		3.6		mA	
	Accelerometer only (DMP & Gyroscope disabled)		500		µA	
Accelerometer Low Power Mode Current	1 Hz update rate		10		µA	
	5 Hz update rate		20		µA	
	20 Hz update rate		70		µA	
	40 Hz update rate		140		µA	

KUVA 25. Gyrokoopin ja kiihtyvyyssanturin virrankulutus (2, s. 14)

µOLED-96-G1-näyttömoduulin virrankulutus kuvan 26 perusteella oli minimissään 14 mA, ja tyypillisesti 40 mA, sekä maksimissaan 120 mA.

Global Characteristics based on Operating Conditions					
Parameter	Conditions	Min	Typ	Max	Units
Supply Current (ICC)	VCC = 5.0V	14	40	120	mA
Output Low Voltage (VOL)	TX pin, IOL = 3.4mA	--	--	0.4	V
Output High Voltage (VOH)	TX pin, IOL = -2.0mA	2.4	--	3.3	V
A/D Converter Resolution	SWITCH pin	--	8	--	bits
Capacitive Loading	All pins	--	--	50	pF
Flash Memory Endurance	GOLDELOX-SGC PmmC Programming	--	1000	--	E/W

KUVA 26. μ OLED-96-G1-näyttömoduulin virrankulutus (4, s. 14)

Ramtrom FM25V05 FRAM -virrankulutus valmiustilassa oli kuvan 27 perusteella 90 μ A ja lepotilassa 5 μ A.

Low Voltage, Low Power

- Low Voltage Operation 2.0V – 3.6V
- 90 μ A Standby Current (typ.)
- 5 μ A Sleep Mode Current (typ.)

KUVA 27. Ramtrom FM25V05 FRAM -lisämuistin virrankulutus (6, s. 1)

Laitteistojen virrankulutuksien perusteella yhteiseksi virrankulutukseksi laitteistolle saatiin noin 64,9 mA. Virtalähteen kapasiteetiksi haluttiin siis vähintään 200 mAh, jotta laitteisto pysyisi langattomasti ainakin kaksi tuntia päällä testikäytössä.

Atmega1280 mikrokontrolleri toimi 2,8-5,5 V:n käyttöjännitteellä, MPU-6000-mikropiiri toimi noin 2,3-3,4 käyttöjännitteellä ja Ramtrom FM25V05 FRAM -lisämuisti toimi 2,0-3,6 V:n käyttöjännitteellä. Lisäksi μ OLED-96-G1-näyttömoduuli tarvitsi 4,0-5,5 V:n käyttöjännitteen. Valitun virtalähteen minimijännitteeksi valittiin siis näyttömoduulin tarvitsema 4,0 V:a. Virtalähteet,

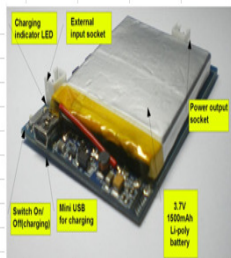





joiden jännitteeksi ilmoitettiin vähemmän kuin 4,0 V:a, jouduttiin ottamaan käyttöön kuvan 28 mukainen DC-DC-konvertteri. Konvertteri nostaa 1,0 - 4,0 V:n sisääntulojännitteen 5,0 V:iin.



KUVA 28. 5 V DC-DC-konvertteri (8)

Kuvaan 29 koottiin kahdeksan erilaista virtalähteratkaisua ja virtalähdevaihtoehdot numeroitiin järjestyksessä. Kuvan 29 punainen väri osoittaa virtalähteen huonoa ominaisuutta, ja vihreä hyvää ominaisuutta. Keltainen väri osoittaa virtalähteelle sitä, että myöhemmässä vaiheessa kyseinen ominaisuus voi olla riski. Ensimmäiseksi vaihtoehdoksi otettiin (shield step-up board), johon oli liitetty DC-DC-teholähde ja 3,7 V:n litiumpolymeeriakku, sekä akun kapasiteetti oli 1500 mAh. Ensimmäinen vaihtoehto oli kooltaan liian suuri, eikä näin ollen soveltunut mitoiltaan rannetietokoneeseen. Vaihtoehdot 2–4 valittiin vertailtavaksi siten, että niistä jokainen oli saman valmistajan virtalähteitä, mutta erodivat toisistaan kooltaan, kapasiteetiltaan ja hinnaltaan. Toiseksi vaihtoehdoksi valittiin litium-ioni polymeeriakku 3,7 V:n jännitteellä ja 1000 mAh kapasiteetilla. Kolmanneksi vaihtoehdoksi valittiin 3,7 V:n ja 400 mAh kapasiteetin omaava litium-ionipolymeeriakku. Neljänneksi vaihtoehdoksi valittiin 3,7 V:n ja 110 mAh kapasiteetin litium-ioni polymeeriakku. Laitteiston virran kulutuksen takia 110 mAh kapasiteetti jäi liian alhaiseksi, joten neljäs vaihtoehto ei soveltunut rannetietokoneeseen. Seuraavaksi vertailtavaksi virtalähteeksi valittiin 7,2 V:n ja 1000 mAh kapasiteetin virtalähde. Virtalähteen mitat ovat melko suuret, joten tämä virtalähde otettiin tässä vaiheessa vertailua pois. Kuudenneksi

virtalähteeksi valittiin 11,1 V:n ja 800 mAh kapasiteetin virtalähde. Tämän virtalähteen korkean hinnan takia tämäkin virtalähde otettiin vertailusta tässä vaiheessa pois. Seuraava virtalähde otettiin vertailuun pienen kokonsa ansiosta. Kyseisen virtalähteen kapasiteetiksi luvattiin 100 mAh ja jännitteeksi 3,0 V:a. Kuitenkin virtalähde otettiin vertailusta tässä vaiheessa pois korkean hinnan ja pienen kapasiteetin takia. Viimeiseksi virtalähteeksi valittiin 12,0 V:n ja 50 mAh kapasiteetin virtalähde. Virtalähde otettiin vertailusta tässä vaiheessa pois, koska jännite oli turhan suuri ja 50 mAh kapasiteetti liian pieni.

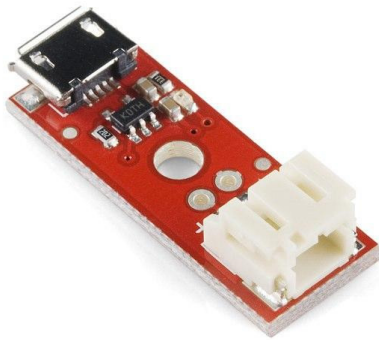
<p>1.</p>  <p>5V DC-DC shield step-up board 3.7V 1500mAh Li-Poly battery Dimensions: 78.7mm x 56.1mm Can be charged from on-board mini-USB socket panel or from external source, such as solar Price £15.50 http://www.nuelectronics.com</p>	<p>2.</p>  <p>Polymer Lithium Ion Battery 3.7V / 1000mAh Dimensions: (53 x 33 x 5.7 mm) Price 11,95 € http://www.sparkfun.com/products/339</p>	<p>3.</p>  <p>Polymer Lithium Ion Battery 3.7V 400mAh Dimensions: (5 x 25 x 35mm) Price 7,95 http://www.sparkfun.com/products/10718</p>	<p>4.</p>  <p>Polymer Lithium Ion Battery 3.7V 110mAh Dimensions: (5.7x12x28mm) Price 6,95 http://www.sparkfun.com/products/731</p>																		
<p>5.</p>  <p>Polymer Lithium Ion Battery 7.4V 2-cell pack / 1000mAh Dimensions: 70mm x 35mm x 18mm Price \$6.95 This battery need a compatible charger http://www.sparkfun.com/</p>	<p>6.</p>  <p>LiPo-battery 11.1V / 800mAh Dimensions: 56 x 31 x 25 mm Charge voltage 12.6 V Price 20 € www.eifa.se</p>	<p>7. SANYO Lithium</p> <p>Cell Type ML2430</p> <p>Specifications</p> <table> <tr><td>Nominal Capacity¹</td><td>100mAh</td></tr> <tr><td>Nominal Voltage</td><td>3V</td></tr> <tr><td>Standard Charge/Discharge Current</td><td>0.5mA</td></tr> <tr><td>Max. Discharge Current</td><td>Continuous² 10mA Pulse² 20mA</td></tr> <tr><td>Charge/Discharge Cycle Characteristics</td><td>Discharging Depth of 5% 3000 Discharging Depth of 20% 500</td></tr> <tr><td>Charging Method</td><td>Constant Voltage Charge 3.140-3.15V High Temperature 2.95±0.15V</td></tr> <tr><td>Temperature Range</td><td>-20°C~+60°C</td></tr> <tr><td>Weight</td><td>4.1g</td></tr> <tr><td>Dimensions</td><td>Diameter(D) 24.5mm Height(H) 3.0mm d 20mm</td></tr> </table> <p>¹ Nominal capacity is determined to an end voltage of 2.0V when the battery is allowed to discharge at a standard current level at 23°C. ² Current value is determined so that 50% of the nominal capacity is obtained with an end voltage of 2.0V at 23°C.</p> <p>Sanyo Ladattava nappiparisto http://www.parcio.biz/verkkokauppa/product_info.php?cPath=15_1113_1472&products_id=9780 Price 23,95 € 3 V / 100 mAh</p>	Nominal Capacity ¹	100mAh	Nominal Voltage	3V	Standard Charge/Discharge Current	0.5mA	Max. Discharge Current	Continuous ² 10mA Pulse ² 20mA	Charge/Discharge Cycle Characteristics	Discharging Depth of 5% 3000 Discharging Depth of 20% 500	Charging Method	Constant Voltage Charge 3.140-3.15V High Temperature 2.95±0.15V	Temperature Range	-20°C~+60°C	Weight	4.1g	Dimensions	Diameter(D) 24.5mm Height(H) 3.0mm d 20mm	
Nominal Capacity ¹	100mAh																				
Nominal Voltage	3V																				
Standard Charge/Discharge Current	0.5mA																				
Max. Discharge Current	Continuous ² 10mA Pulse ² 20mA																				
Charge/Discharge Cycle Characteristics	Discharging Depth of 5% 3000 Discharging Depth of 20% 500																				
Charging Method	Constant Voltage Charge 3.140-3.15V High Temperature 2.95±0.15V																				
Temperature Range	-20°C~+60°C																				
Weight	4.1g																				
Dimensions	Diameter(D) 24.5mm Height(H) 3.0mm d 20mm																				

KUVA 29. Virtalähde vaihtoehdot

5.2 Valittu virtalähderatkaisu

Virtalähteeksi valittiin kolmantena oleva 3,7 V:n ja 400 mAh virtalähde. Virtalähteen kapasiteetti oli riittävän suuri ja jännite voitiin nostaa DC-DC-konvertterin avulla 5,0 V:iin. Virtalähde oli hinnaltaan sopiva, ja sitä voitiin ladata

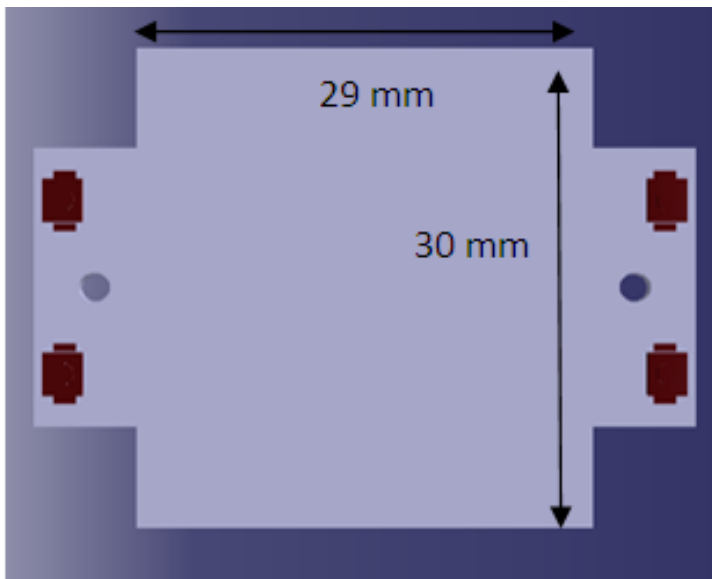
kuvan 30 SparkFun litiumpolymeeri-akkulaturilla. Akkulaturin syöttöjännite virtalähteelle oli 3,7 V:a, ja sillä voitiin ladata 500 mAh kapasiteetti täyteen tunnissa.



KUVA 30. SparkFun litiumpolymeeri akkulaturi (9)

5.3 Virtalähteen edut ja haitat

Virtalähteen etuina olivat pienet mitat 5,7 * 12 * 28 mm, ja riittävän suuri kapasiteetti. Pienien mittojen ansiosta virtalähde saatiin sopimaan rannetietokoneen mekaniikkaan. Rannetietokoneen pohjalevy, johon piirilevy ja komponentit sijoitettiin näkyy kuvassa 31. Pohjalevyn mitat olivat 29 * 30 mm eli virtalähde saatiin sopimaan pohjalevylle. Lisäksi virtalähteen kustannukset olivat pienet, ja sen latautuminen SparkFun litiumpolymeeri-akkulaturilla täyteen tapahtui alle tunnissa.



KUVA 31. Rannetietokoneen pohjalevy (10, s. 16)

Virtalähteen haittapuoleksi osoittautui se, että 400 mAh kapasiteetti ei välttämättä ole tarpeeksi paljon rannetietokoneelle. Toisena haittapuolena oli, että rannetietokoneeseen jouduttiin integroimaan ylimääräistä latauselektroniikkaa, jotta virtalähdettä voitaisiin ladata uudelleen. Lisäksi elektroniikkaa jouduttiin lisäämään virtalähteen takia, koska virtalähteen 3,7 V:n jännite ei riittänyt elektroniikka-alustalle, vaan jouduttiin integroimaan DC-DC-konvertteri.

6 YHTEENVETO

Opinnäytetyössä kehitettiin toisen sukupolven SpoCo-projektiin kuuluvaa rannetietokonetta. Rannetietokoneen laitteistonsuunnittelu ja ohjelmistonkehitys tapahtui projektin vaatimuksien mukaisesti. Rannetietokoneen laitteiston integroinnin kehityksessä saatiin aikaiseksi laite, jolla voitaisiin testata sellaisia ominaisuuksia, jotka tulisivat seuraavaan ranteeseen asennettavaan pienikokoisempaan prototyyppiin. Projektissa laitteiston, mekaniikan, virrankulutuksen ja ohjelmiston osalta ei haettu tarkkoja vaatimuksia. Projektissa syntyneellä laitteella tapahtuisi jatkossa seuraavan prototyypin spesifikaatioiden määrittely. Seuraavan prototyypin laitteistonsuunnittelu ja ohjelmistokehitys vaatii paljon enemmän resursseja ja suunnitelmallisuutta, koska seuraavan prototyypin vaatimuksissa on, että rannetietokoneella pitää voida asentaa käyttäjän ranteeseen. Lisäksi seuraavassa prototyypissä tulee olla vähintään samat ominaisuudet kuin toisen sukupolven rannetietokoneessa. Ominaisuuksien tulee myös toimia rannetietokoneessa luotettavasti. Lisäksi seuraavan prototyypin laitteistolta vaaditaan parempaa virrankulutusta, ja laitteiston tulee olla integroitu mahdollisimman pienelle piirilevylle.

Opinnäytetyön yhteydessä vertailtiin eri valmistajien virtalähteitä keskenään. Valittu virtalähde tulee projektin rannetietokoneen virtalähteeksi, ja sen kautta voidaan lähteä kartoittamaan tarkemmin seuraavan prototyypin tarvitsemaa virtalähdettä. Lisäksi projektin aikana käytiin läpi valitun virtalähteratkaisun etuja ja haittoja.

Projektin aikana saatiin hyvin integroitua laitteisto yhteen laitteisto- ja ohjelmisto tasoilla. Tästä johtuen tulevaisuudessa varmasti otetaan entuudestaan tutut laitteistot mukaan seuraavaan prototyyppiin. Seuraava vaihe on saada laitteisto yhdelle ja samalle piirilevylle, jossa on otettu huomioon myös mekaaniset vaatimukset.

Projektiin lähdin mukaan marraskuussa 2011, ja projekti todella osoittautui haastavaksi ja välillä projekti tuntui erittäin laajalta. Minulla ei ollut aikaisempaa

kokemusta näin laajan laitteistokokonaisuuden ja ohjelmistonkehityksen kehittämistä. Projektiin liittyvät aihealueet eivät kuitenkaan olleet vieraita minulle, koska harrastan vapaa-ajallani elektroniikkaa, ja koulutukseni suuntautuminen vahvisti osaamista projektissa. Lisäksi haastellisuutta lisäsi, että projektini kehityksessä tuli ottaa huomioon se, että lopullinen tuote tulisi lanseeraamaan markkinoille. Laitteiston ja ohjelmiston suunnittelu oli minulle kiinnostavaa, ja erittäin hyvän kokemuksen antoivat myös opinnäytetyön aikana pidetyt tiimipalaverit. Tulevaisuudessa tarvitaan kuitenkin paljon tarkempaa suunnittelua, ja enemmän aktiivisuutta ryhmätyöskentelyyn.

LÄHTEET

1. Arduino ATmega1280. 2012. Saatavissa:
<http://arduino.cc/en/Main/ArduinoBoardMega>. Hakupäivä 24.4.2012.
2. InvenSense Inc. 2011. MPU-6000 and MPU-6050 Product Specification Revision 3.2. Saatavissa:
<http://www.invensense.com/mems/gyro/documents/PS-MPU-6000A.pdf>.
Hakupäivä 24.4.2012.
3. SpoCo. 2011. SpoCo-Bluetooth connection. Saatavissa:
<https://optima.oamk.fi/learning/id6/bin/user?rand=8256>. Hakupäivä 2.5.2012.
4. 4D SYSTEMS. 2011. µOLED-96-G1(SGC) Serial OLED Display Module Data Sheet. Saatavissa: <http://www.4dsystems.com.au/downloads/Serial-Display-Modules/uOLED-96-G1%28SGC%29/Docs/uOLED-96-G1SGC-DS-rev6.pdf>. Hakupäivä: 2.5.2012
5. RobotShop. 2012. 4D Systems 0.96" OLED Display Shield for Arduino. Saatavissa: <http://www.robotshop.com/ProductInfo.aspx?pc=RB-Fds-20>.
Hakupäivä: 2.5.2012
6. RAMTRON. 2012. FM25V05 512Kb Serial 3V F-RAM Memory. Saatavissa:
http://www.ramtron.com/files/datasheets/FM25V05_ds.pdf. Hakupäivä
20.5.2012.
7. ATMEL. 2012. 8-bit Atmel Microcontroller with 64K/128K/256K Bytes In-System Programmable Flash. Saatavissa:
<http://www.atmel.com/Images/doc2549.pdf>. Hakupäivä 20.5.2012.
8. SparkFun. 2007. NCP1402-5V Step-Up Breakout. Saatavissa:
<http://dlnmh9ip6v2uc.cloudfront.net/images/products/10968-01.jpg>.
Hakupäivä: 28.5.2012.

9. SparkFun. 2008. LiPo Charger Basic - Micro-USB. Saatavissa: <http://dlnmh9ip6v2uc.cloudfront.net/images/products/10217-01.jpg>.
Hakupäivä: 28.5.2012.
10. Renaudin, Hugo. 2011. Mechanical desing of the Spoco watch. Oulu University of applied science. Saatavissa: [https://optima.oamk.fi/learning/id6/bin/doc_show?id=59813&ws=12903&noedit=1&name=/report\(2\).docx](https://optima.oamk.fi/learning/id6/bin/doc_show?id=59813&ws=12903&noedit=1&name=/report(2).docx). Hakupäivä: 28.5.2012.
11. Alakärppä, Juha. 2012. SA/SD-RT –mallintamisesimerkki. Saatavissa: http://www.tekniikka.oamk.fi/~juhaalak/.sulohj/RealjOhj/SA_SD_RT_Example.pdf. Hakupäivä: 27.5.2012.

LIITTEET

Liite 1. (OAMK:n tekniikan yksikön käytössä)

Liite 2. (OAMK:n tekniikan yksikön käytössä)

Liite 3. (OAMK:n tekniikan yksikön käytössä)

Liite 4. (OAMK:n tekniikan yksikön käytössä)

Liite 5. (OAMK:n tekniikan yksikön käytössä)